Experimental Results: Random digits and hypothesis testing

---

48 marks

In this question, we will explore the data collected in class.

You will need to download the data `class_data.csv` from the assignment website and save it somewhere. Supposing the data to have been saved in a directory `dataDirectory`, read it into R as

```r
# Usual helper for paths
path_concat <- function(path1, path2, sep="/") {
                        paste(path1, path2, sep = sep)
}

data <- read.csv(file = path_concat(dataDirectory, "class_data.csv"))
```

Having loaded the data, you might have a look at its contents using any of the standard functions:

```r
# just print it
data
# view its contents as a spreadsheet (in RStudio)
View(data)
# or, look at its data structure
str(data)
```

You will find that it is a `data.frame` with variables

```r
names(data)
```

```
## [1] "random_digit"  "student_digit" "green_card1"   "green_card2"
## [5] "red_card1"     "red_card2"
```

Each row contains one student's answer to the questions associated with these names.

Here we will only consider the results on the digits (0-9) obtained in class.

Recall how the data on the digits 0, 1, …, 9 were collected.

Each person was first asked to write the words "random digit" on a card. They were then given a few seconds to think up a single *random* digit from 0 to 9 and then record it on the card.

On the other side of the card, each person then wrote "student digit" and below it recorded the *last* digit of their student id number.

These two digits provide the values for the variables `random_digit` and `student_digit` appearing in the data set `data` above.

---

1. Suppose a digit, d, is generated as a realization from a random variable D which is uniformly distributed on the digits $\{0, 1, 2, 3, \ldots, 8, 9\}$. That is, for any value $d \in \{0, 1, 2, 3, \ldots, 8, 9\}$ we have

$$Pr(D = d) = \frac{1}{10}.$$

a. *(1 mark)* Determine the expectation $E(D)$.

$$E(D) = \sum_{D=0}^{9} D \cdot P(D) = \sum_{D=0}^{9} \frac{D}{10} = 4.5$$

b. *(2 marks)* Determine the expectation $E(D^2)$ and hence the standard deviation $SD(D)$.

$$E(D) = \sum_{D=0}^{9} D^2 \cdot P(D) = \sum_{D=0}^{9} \frac{D^2}{10} = 28.5$$

$$Var(D) = E(D^2) - E(D)^2 = 28.5 - 4.5^2 = 8.25$$

$$SD(D) = \sqrt{(}Var(D)) = \sqrt{8.25} = 2.872281$$

c. *(1 mark)* Determine the median of D.

$$median(D) = 4.5$$

d. Suppose we consider the random variable C which for some fixed value $d \in \{0, 1, 2, 3, \ldots, 8

$$C = \begin{cases} 1 & \text{when } D = d \\ 0 & \text{when } D \neq d \end{cases}$$

which implies

$$Pr(C = 1) = Pr(D = d) = \frac{1}{10}$$

and

$$Pr(C = 0) = Pr(D \neq d) = 1 - Pr(D = d) = \frac{9}{10}$$

Suppose we have $C_1, C_2, \ldots, C_n$ independent and identically distributed random variables wi

Let $X = \sum_{i=1}^n C_i$.

i. *(1 mark)* What is the name of the probability distribution of X?
- binomial distribution

ii. *(1 mark)* Hence, write down an expression $Pr(X = x)$ for $x \in {0, 1, \ldots, n}$.

$$Pr(X = x) = \binom{n}{k} \left(\frac{1}{10}\right)^k \left(1 - \frac{1}{10}\right)^{n-k}$$

```
    iii. *(1 mark)* Hence, write down an expression $E(X)$.
```

$$E\left(X\right) = np = \frac{n}{10}$$

```
e. Using the `data` from class, for **each** of the variables `random_digit` and `student_digit`, calcul

    i. *(2 marks)* sample average,
```

```r
s_digit <- data$student_digit
r_digit <- data$random_digit
s_digit_mean <- mean(s_digit)
r_digit_mean <- mean(r_digit)
print(s_digit_mean)
```

```
## [1] 4.595238
```

```r
print(r_digit_mean)
```

```
## [1] 5.5
```

```
        - The mean of student_digit is 4.595238
        - The mean of random_digit is 5.5
```

```
    ii. *(2 marks)* sample standard deviation,
```

```r
s_digit_sd <- sd(s_digit)
r_digit_sd <- sd(r_digit)
print(s_digit_sd)
```

```
## [1] 3.052864
```

```r
print(r_digit_sd)
```

```
## [1] 2.778401
```

```
        - The standard deviation of student_digit is 3.052864
        - The standard deviation of random_digit is 2.778401
```

```
    iii. *(2 marks)* sample median,
```

```r
s_digit_median <- median(s_digit)
r_digit_median <- median(r_digit)
print(s_digit_median)
```

```
## [1] 5
```

```r
print(r_digit_median)
```

```
## [1] 7
```

```
        - The median of student_digit is 5
        - The median of random_digit is 7
```

iv. *(3 marks)* and compare these to the corresponding theoretical values from the distribution of S

Varname	Theoretical	student_digit	random_digit
mean	4.5	4.595238	5.5
median	4.5	5	7
standard deviation	2.872281	3.052864	2.778401

* student_digit sample has larger mean, median,
  and standard deviation than theoretical value
* random_digit sample has larger mean,
  and median than theoretical results.
* But its standard deviation is smaller than theoretical value

v. *(1 mark)* Which of `random_digit` and `student_digit` have sample values closer to the theoretic

- student_digit have sample values closer to the theoretical
  values because the median and mean of student_digit
  is closes to theroretical values. student_digit and random_digit
  have standards deviation are all close to the theoretical sd.

-- Student_digit is more close to theoretical(uniform distribution)

f. *(3 marks)* Calculate $Pr(X = x)$ when $n =$ 42 and $x = 0$, 5, and 10.

$$Pr(X = 0) = \binom{42}{0}\left(\frac{1}{10}\right)^{0}\left(\frac{9}{10}\right)^{42-0} = 0.01197251518$$

$$Pr(X = 5) = \binom{42}{5}\left(\frac{1}{10}\right)^{5}\left(\frac{9}{10}\right)^{42-5} = 0.17247769726$$

$$Pr(X = 10) = \binom{42}{10}\left(\frac{1}{10}\right)^{10}\left(\frac{9}{10}\right)^{42-10} = 0.00505246993$$

2. We are interested testing the hypothesis

   H: the observed digits d_1, d_2, ..., d_n are independent realizations of a random variable D uniformly distributed on the digits $\{0, 1, 2, \ldots, 9\}$.

   In particular, we are interested in testing this hypothesis for each of two samples `student_digit` and `random_digit`.

   a. *(4 marks)* The function `stem()` is a simple way to get a quick picture (a "stem and leaf plot") of the distribution of a set of digits. Use `stem()` to construct a picture of each of the following:

      i. the values of `student_digit`,

```r
stem(s_digit)
```

```
##
##   The decimal point is at the |
##
##   0 | 000000
##   1 | 0000
##   2 | 000
##   3 | 0
##   4 | 000000
##   5 | 000
##   6 | 0000000
##   7 | 0
##   8 | 0000000
##   9 | 0000
```

```
    ii. the values of `random_digit`,
```

```r
stem(r_digit)
```

```
##
##   The decimal point is at the |
##
##   0 | 000
##   1 | 0
##   2 | 000
##   3 | 000000
##   4 | 00
##   5 | 00
##   6 | 000
##   7 | 000000000000
##   8 | 0000
##   9 | 000000
```

```
    iii. and, for comparison, a sample of the same size from a uniform distribution on the digits using
        the function `sample()`.
```

```r
uniform_sample <- sample(1:10, replace = TRUE,40)
stem(uniform_sample)
```

```
##
##   The decimal point is at the |
##
##    1 | 0000
##    2 | 00
##    3 | 0000
```

```
##    4 | 000000
##    5 | 000
##    6 | 00000
##    7 | 00
##    8 | 000
##    9 | 0000000
##   10 | 0000
```

    Which of `student_digit` or `random_digit` looks more like it might have come from a Uniform on the

        - student_digit looks more like it might have come from a Uniform on the digits becaus the leng

b. A more formal way to assess whether a sample of values appear to come from a hypothesized distributi

$$\chi^2 = \sum_{i=1}^{m} \frac{(o_i - e_i)^2}{e_i}$$

    where o_i is the observed number of values in the ith "cell", e_i is the expected number to f

    In our case, the cells are the 10 different possible digits (so $m = 10$) and o_i is the number o

    $\Chi^2$ is a discrepancy measure which is larger whenever the o_i are relatively far from their

    If the hypothesized model is true, then the distribution of $\Chi^2$ can usually be approximated a

    i. *(4 marks)* Write a function `count_digits()` which takes a vector of digits `d` and returns a n

```r
count_digits <- function (d) {
  len<- length(d)
  digits <- seq(0,len-1,1)
  return(sapply(digits,function(x) sum(d==x))[1:10])
}



my_digits <- c(0, 1, 3, 4, 7, 1, 4, 9, 7, 4)
print("my_function returns:")
```

```
## [1] "my_function returns:"
```

```r
count_digits(my_digits)
```

```
##  [1] 1 2 0 1 3 0 0 2 0 1
```

```r
# would return a vector equal to
print("it should be: ")
```

```
## [1] "it should be: "
```

```r
c(1, 2, 0, 1, 3, 0, 0, 2, 0, 1)
```

```
##  [1] 1 2 0 1 3 0 0 2 0 1
```

```
        # for example if
        my_digits <- c(0, 1, 3, 4, 7, 1, 4, 9, 7, 4)
        # then
        count_digits(my_digits)
```

```
        # would return a vector equal to
        c(1, 2, 0, 1, 3, 0, 0, 2, 0, 1)
        ```

        Note, that we will assume that `d` will be an integer vector containing
        only values in $\{0, 1, 2, \ldots , 9 \}$.

        No error checking is required for now.

    ii. *(2 marks)* Demonstrate your function on the digits of the variable `student_digit` and on the
```

```r
count_digits(s_digit)
```

```
##  [1] 6 4 3 1 6 3 7 1 7 4
```

```r
count_digits(r_digit)
```

```
##  [1]  3  1  3  6  2  2  3 12  4  6
```

```
    iii. *(4 marks)* Write the function `Pearson_chi_sq(observed, expected)` which calculates $\Chi^2$ v

            Again, for expediency it will be assumed that both `observed` and `expected` vectors contai

            However, you should check that the lengths of `observed` and `expected` match and stop if th
```

```r
Pearson_chi_sq <- function (observed,
                            expected = sum(observed)/length(observed)) {
  if(length(expected) == 1){
    expected = rep(expected,length((observed)))
  }
  if(length(observed) != length(expected)){
    print("Lengths do not match")
    return()
  }
  values <- mapply(function(x,y){(x-y)^2/y}, observed, expected)
  return(sum(values))
}

Pearson_chi_sq(count_digits(data$student_digit))
```

```
## [1] 10.85714
```

```r
Pearson_chi_sq(count_digits(data$random_digit))
```

```
## [1] 21.80952
```

```
        - The Pearson_chi_sq
        -
    iv. *(2 marks)* Check your function by comparing the values of `Pearson_chi_sq(observed)`  to resul

        `count_digits(data$student_digit)` and `count_digits(data$random_digit)` in turn.
```

```r
chisq.test(count_digits(data$student_digit))$statistic
```

```
## Warning in chisq.test(count_digits(data$student_digit)): Chi-squared
## approximation may be incorrect
```
```

```
## X-squared
##  10.85714
```

```r
chisq.test(count_digits(data$random_digit))$statistic
```

```
## Warning in chisq.test(count_digits(data$random_digit)): Chi-squared
## approximation may be incorrect
```

```
## X-squared
##  21.80952
```

```
    v. *(2 marks)* Using the function `pchisq()` calculate the $p$-value testing the uniformity hypothes
    Show that your  -values agree with those produced by chisq.test(observed)$p.value for the counts of
```

```r
pchisq(Pearson_chi_sq(count_digits(data$student_digit)),9,lower.tail = FALSE)
```

```
## [1] 0.2856284
```

```r
pchisq(Pearson_chi_sq(count_digits(data$random_digit)),9,lower.tail = FALSE)
```

```
## [1] 0.009502653
```

```r
chisq.test(count_digits(data$student_digit))$p.value
```

```
## Warning in chisq.test(count_digits(data$student_digit)): Chi-squared
## approximation may be incorrect
```

```
## [1] 0.2856284
```

```r
chisq.test(count_digits(data$random_digit))$p.value
```

```
## Warning in chisq.test(count_digits(data$random_digit)): Chi-squared
## approximation may be incorrect
```

```
## [1] 0.009502653
```

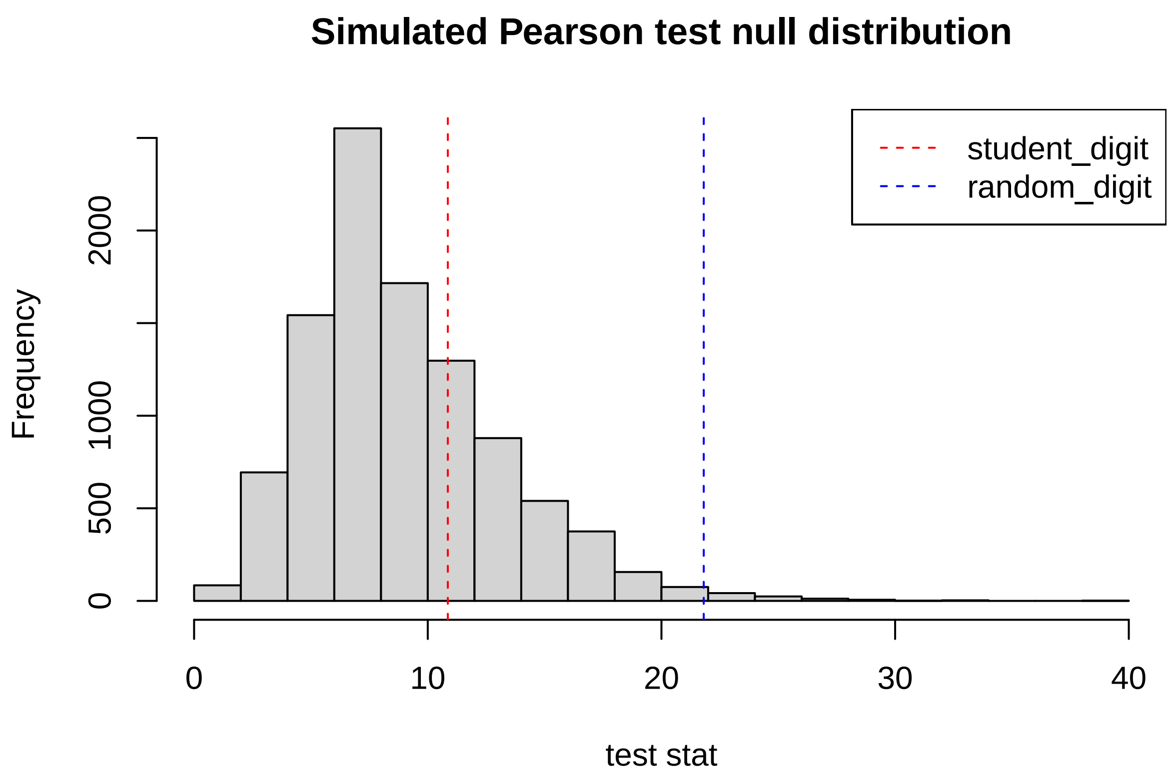
```
    vi. *(3 marks)* Rather than depend upon the validity of the $\chi^2_k$ approximation, we could *simu

        Using the function `sapply()` and the function `sample()`, together with your functions `Pearso

        That is, write
```

```r
get_chisqs <- function (n, B = 1000) {
  count_list <- list()
  for (i in 1:B){
    my_sample <- sample(0:9, n, replace = TRUE)
    count_list[[i]] <- count_digits(my_sample)
  }
  chisq_list<-sapply(1:B, function(x)
  Pearson_chi_sq(count_list[[x]]))
  return(chisq_list)
}

n <- nrow(data)
results <- get_chisqs(n = n, B = 1000)

#print(results)
```

vii. *(3 marks)* Use your function `get_chisqs()` to get `B = 10000` independent pseudo-random real

That is, execute the following (N.B. in RMarkdown change header to eval = TRUE):

```
n <- nrow(data)
B <- 10000 # TEN thousand
set.seed(314159) # So we all get the same values
chisq_stats <- get_chisqs(n = n, B = B)
hist(chisq_stats,
     col = "lightgrey",
     main = "Simulated Pearson test null distribution",
     xlab = "test stat")
student_line <- Pearson_chi_sq(count_digits(data$student_digit))
#print(student_line)
random_line <- Pearson_chi_sq(count_digits(data$random_digit))
#print(random_line)
abline(v=student_line,col='red',lty=2)
abline(v=random_line,col='blue',lty=2)
legend("topright",legend=c('student_digit','random_digit'),col=c('red','blue'),lty =2)
```



To this histogram, add a vertical line in "red" where the corresponding statistics you calcu

Put a legend in the top right that identifies the lines.

Based on this histogram, which collection of digits seem less likely to have been generated
    - Random digit is lesslikely to have been generated as a random sample.
    - The larger is X2, the greater is the evidence against the model.
    - random_digit has large X2 value.

```
    viii. *(2 marks)* The simulated distributions can be used to calculate approximate $p$-values by si

            Calculate these two $p$-values using the simulated test null distribution given by the vect
```

```r
chi_sq_student <- Pearson_chi_sq(count_digits(data$student_digit))
p_value_student <- sum(chisq_stats >= chi_sq_student)/length(chisq_stats)
print(p_value_student)
```

```
## [1] 0.3013
```

```r
chi_sq_random <- Pearson_chi_sq(count_digits(data$random_digit))
p_value_random <- sum(chisq_stats >= chi_sq_random)/length(chisq_stats)
print(p_value_random)
```

```
## [1] 0.0107
```

```
    ix. *(2 marks)*   What do you conclude about the two hypotheses?

      - For student digit, the p-value > 0.05, which means that the null hypothesis can be accepted.
      - For random digit, the p-value < 0.05, which means that the nulll hypothesis should be rejected.
      - student_digit is randomly selected, the random_digit is NOT randomly selected.
```